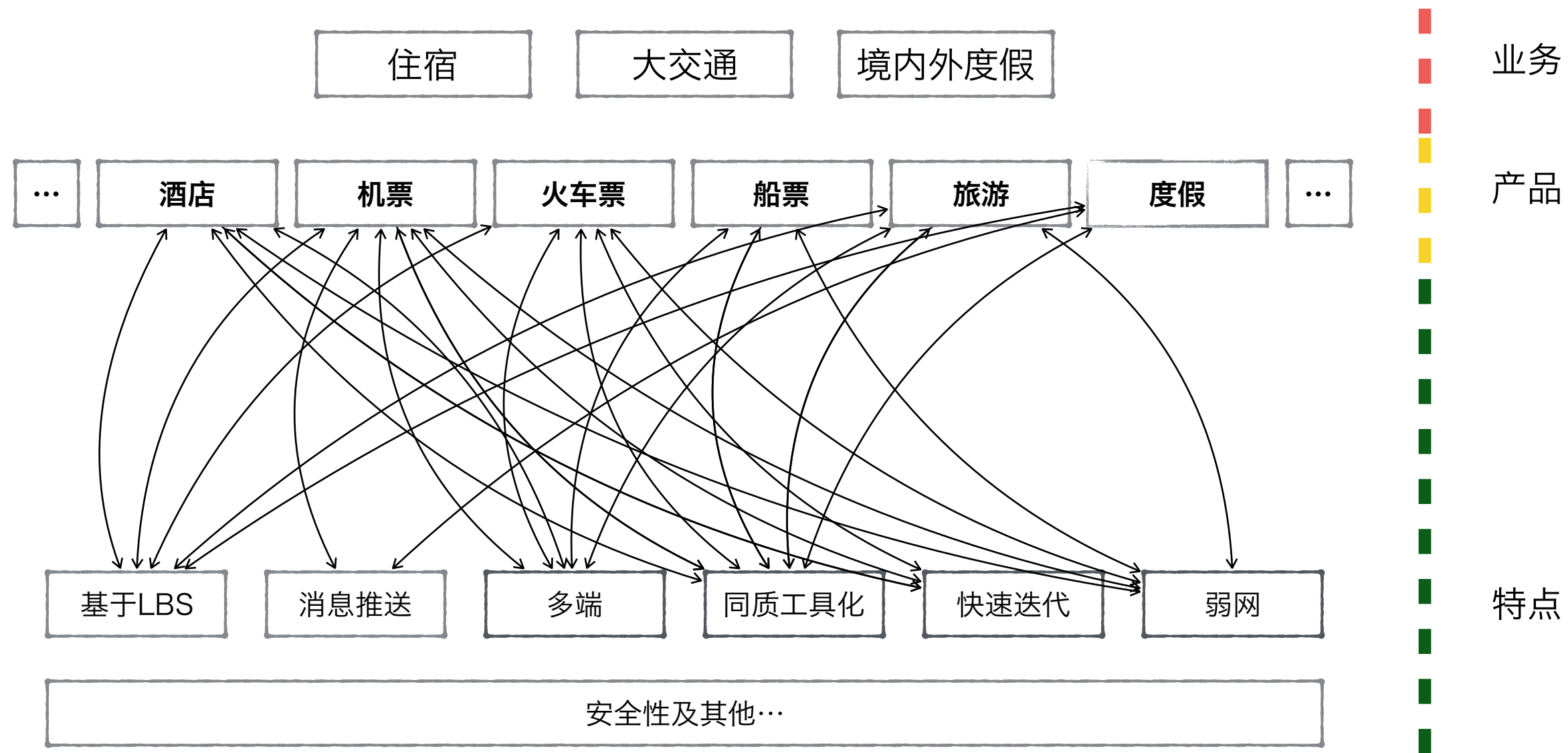


酒旅 Hybrid 架构体系及演进

马荃 @ 新美大酒旅前端 20161017



酒旅



如何做一个好的架构满足业务的发展和成长？

- H5 的快速迭代试错的业务能力 vs. Native 稳定的发版周期：节奏快慢如何取舍？
- H5 的跨平台 vs. Native 单一系统支持：开发资源（人力成本）如何控制？
- H5 的低性能 和 Native 的极致体验：如何兼顾？

| | Native | H5 | 说明 |
|-------------|--------|----|------------------------------|
| 人效（人力成本） | | √ | H5 页面开发不需要单独开发 iOS 和 Android |
| 入口（引流） | | √ | H5 可以跨平台投放、内嵌至各种APP |
| 快速迭代 | | √ | H5 灵活发版，上线无需审核 |
| 跨平台 | | √ | H5 跨平台，运行环境不区分操作系统 |
| 更新 | | √ | H5 可以静默更新不需要应用商店推送更省流量 |
| 性能 | √ | | Native 性能体验很棒，不局限在容器内，渲染更快 |
| 基于LBS（设备调用） | √ | | Native 可以与通过 API 直接调用外置设备 |
| 安全 | √ | | Native 是安全沙箱 |

虽然 H5 部署灵活、快速迭代、方案试错、引流冲量、跨平台、高效率、低成本、三观正，但是：

- 性能被 Native 完爆
- 体验被 Native 完爆

I now pronounce you
hybrid mobile apps



Mobile
Web
Apps



Native
Mobile
Apps

本次分享主要内容：

- 介绍酒旅的 Hybrid 架构在演进的过程中解决了哪些问题
- 对 Hybrid 架构的思考以及展望
- Q&A：顺带招人

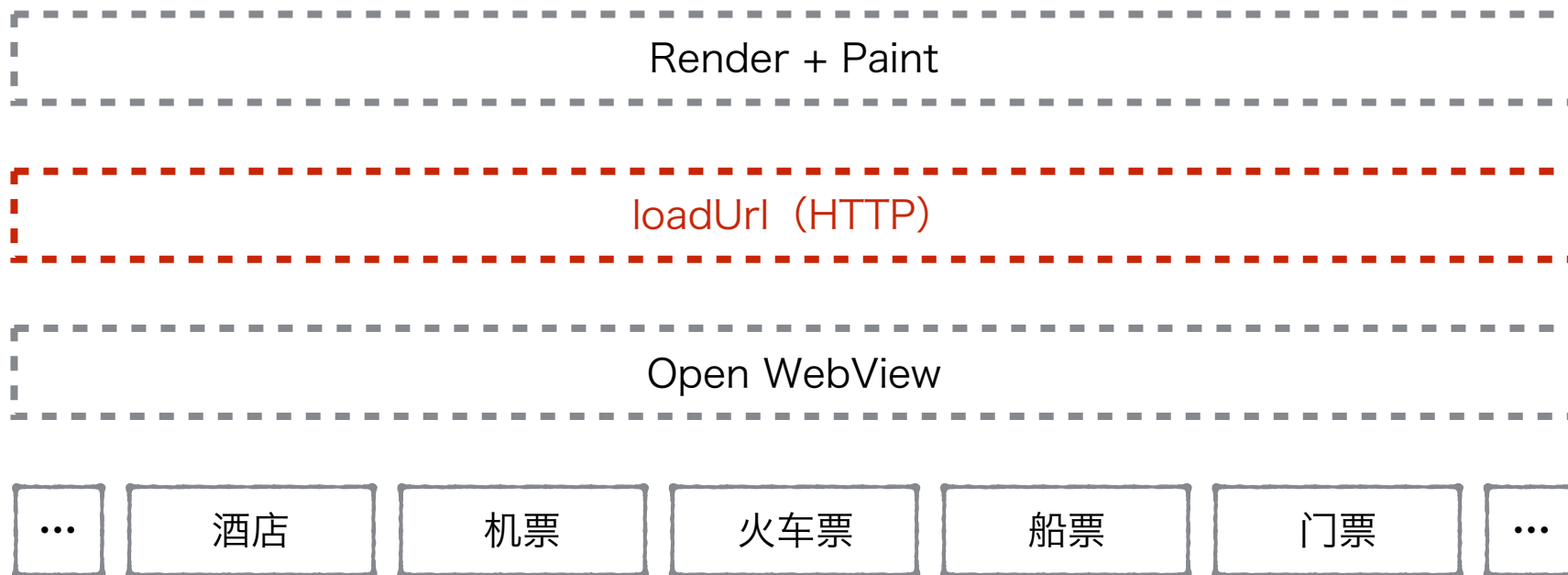
酒旅 Hybrid 架构的前世

- 简单封装了 Native 的能力，但是性能还是 H5 原来的性能
- 新业务接入，各类业务技术栈和选型形态各异，对接后体验效果参差不齐

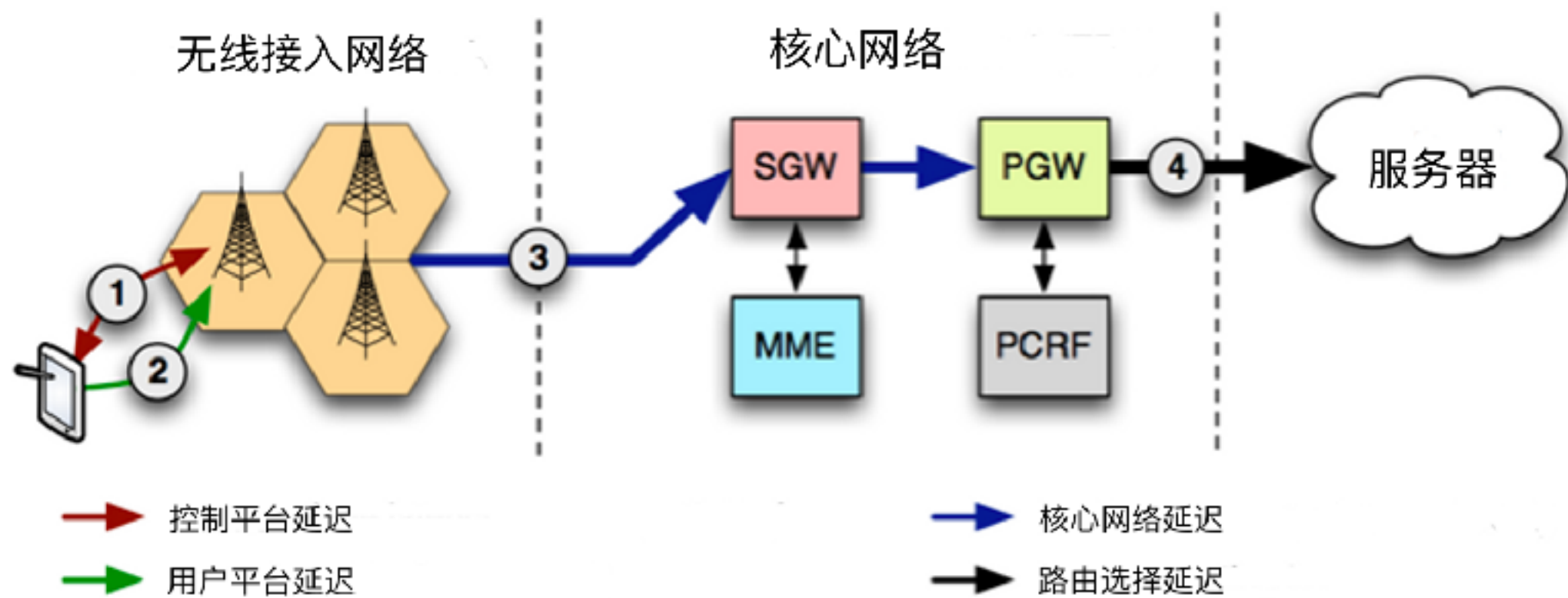
| | Hybrid | 说明 |
|-------------|--------|------------------------------|
| 人效（人力成本） | √ | H5 页面开发不需要单独开发 iOS 和 Android |
| 入口（引流） | √ | H5 可以跨平台投放、内嵌至各种APP |
| 快速迭代 | √ | H5 灵活发版，上线无需审核 |
| 跨平台 | √ | H5 跨平台，运行环境不区分操作系统 |
| 更新 | √ | H5 可以静默更新不需要应用商店推送更省流量 |
| 性能 | × | 简单的融合无法解决 H5 的性能问题 |
| 基于LBS（设备调用） | √ | Native 可以与通过 API 直接调用外置设备 |
| 安全 | √ | Native 是安全沙箱 |

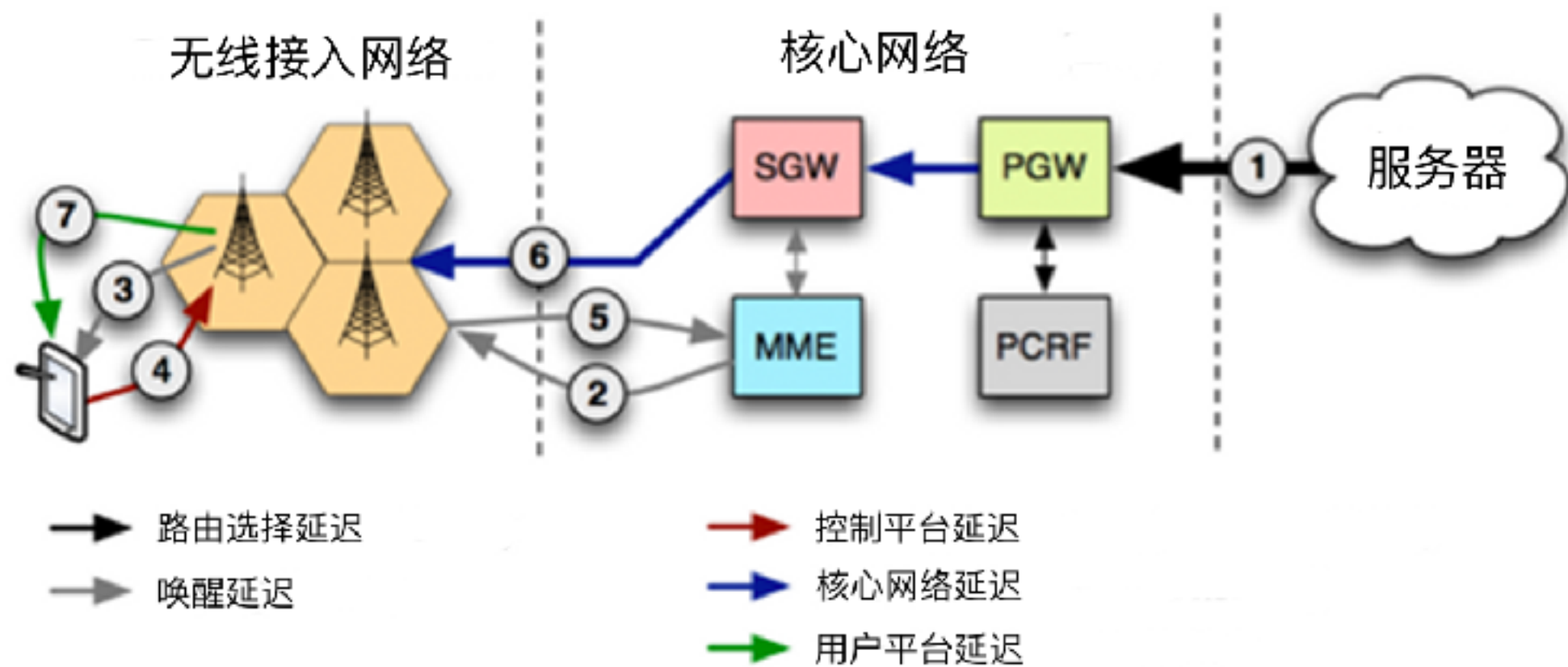
性能问题有哪些？

- WebView 带来的白屏
- 弱网带来的白屏
- 渲染带来的白屏

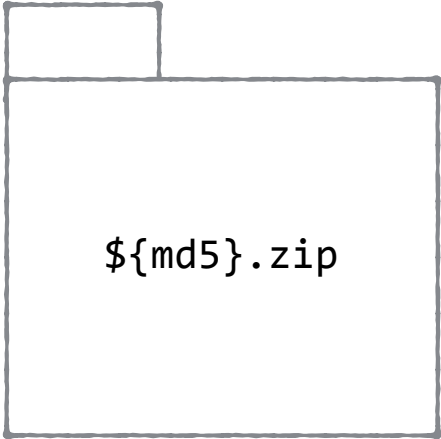
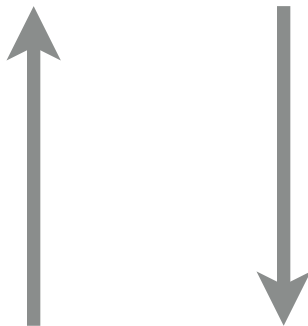


| | 3G 网络 | 4G 网络 |
|----------|---------------|--------------|
| RRC 控制平台 | (200-2500 ms) | (50-100 ms) |
| 域名解析 | 200 ms | 80 ms |
| TCP 连接 | 200 ms | 80 ms |
| TLS 连接 | (200-400 ms) | (80-160 ms) |
| HTTP 连接 | 200 ms | 80 ms |
| 网络延迟 | 600 - 3500 ms | 240 - 500 ms |





Config File Management Sys.



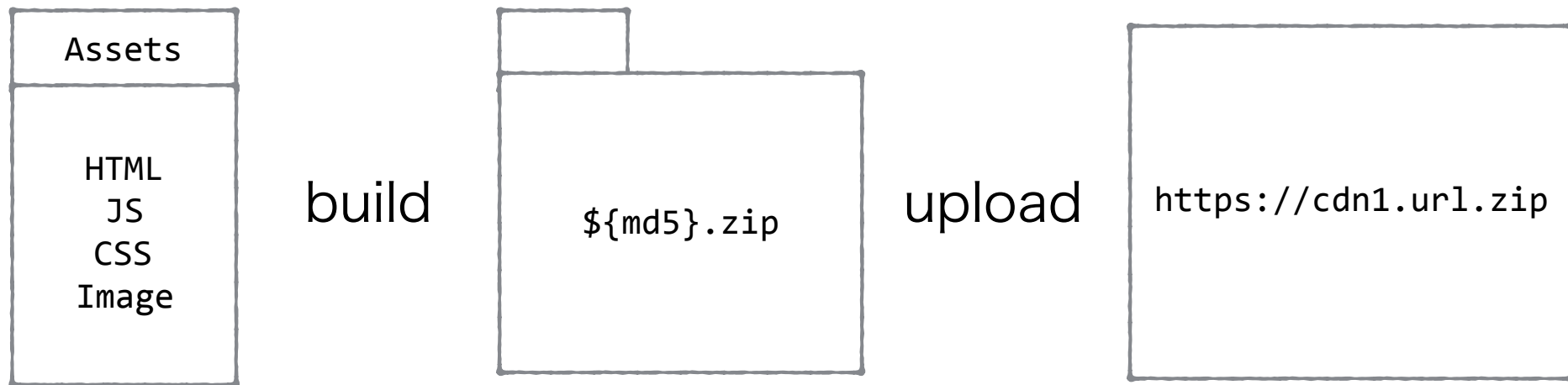



```
$ gulp server
```



```
$ gulp zip
```

```
$ gulp build
```



Config File Management Sys.

```
{
  "data": [
    {
      "name": "xxx",
      "version": "1.0.0",
      "src": "https://cdn1.url.zip",
      "useOfflineZip": true,
      "zipVersion": 000,
      "offlinePreUrl": "https://kyfw.12306.cn/"
    },
    {
      "name": "yyy",
      "version": "1.0.0",
      "src": "https://cdn2.url.zip",
      "useOfflineZip": true,
      "zipVersion": 111,
      "offlinePreUrl": "http://mc.meituan.net/hotel/"
    }
  ]
}
```

local assets > build > local.zip > https://cdn1.url.zip

Config File Management Sys.



Interceptor

Interceptor

Correct. - local

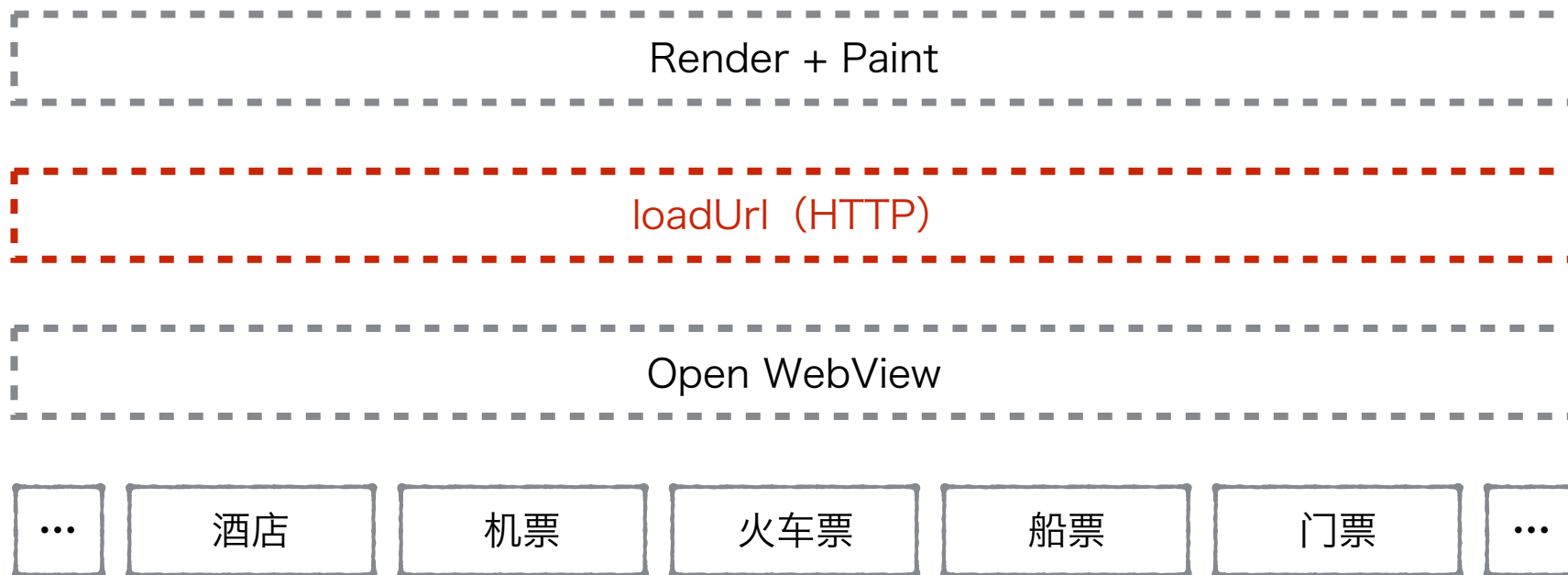
error. - new request

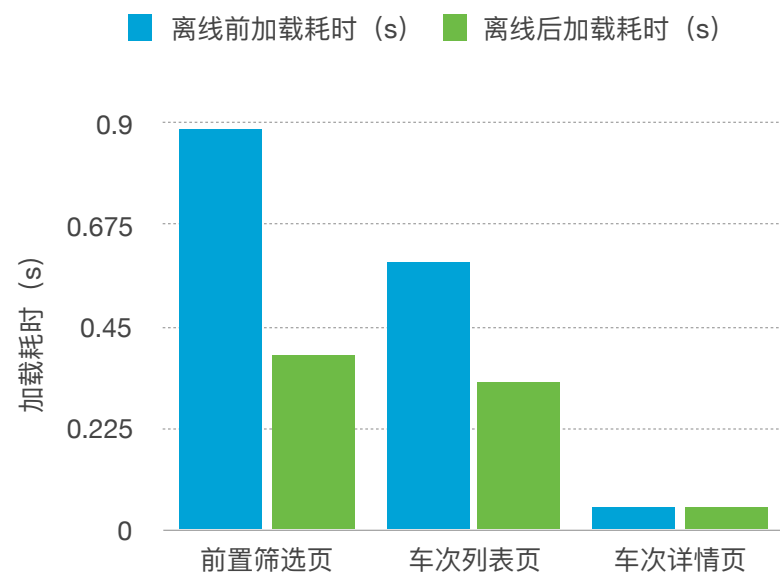
INTERCEPT

mc.meituan.net

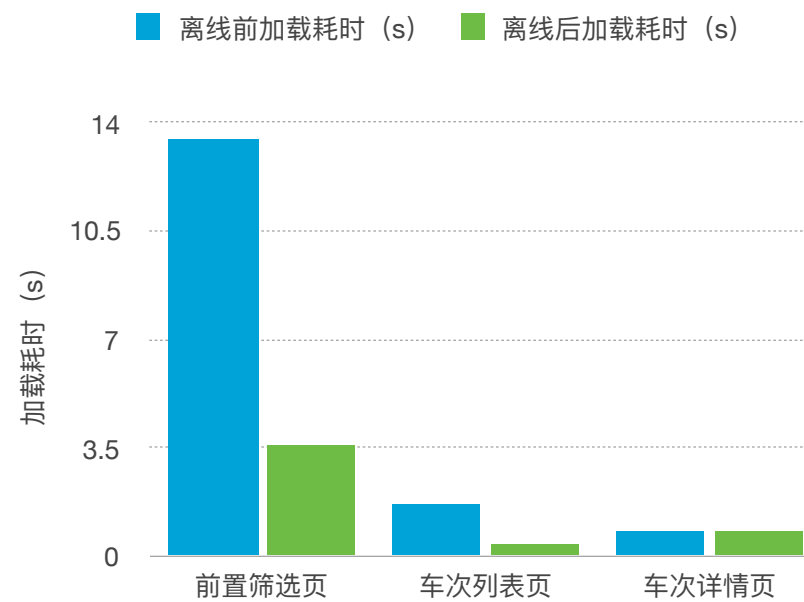
/hotel/common/jquery/dist/jquery.min.js

http://mc.meituan.net/hotel/common/jquery/dist/jquery.min.js

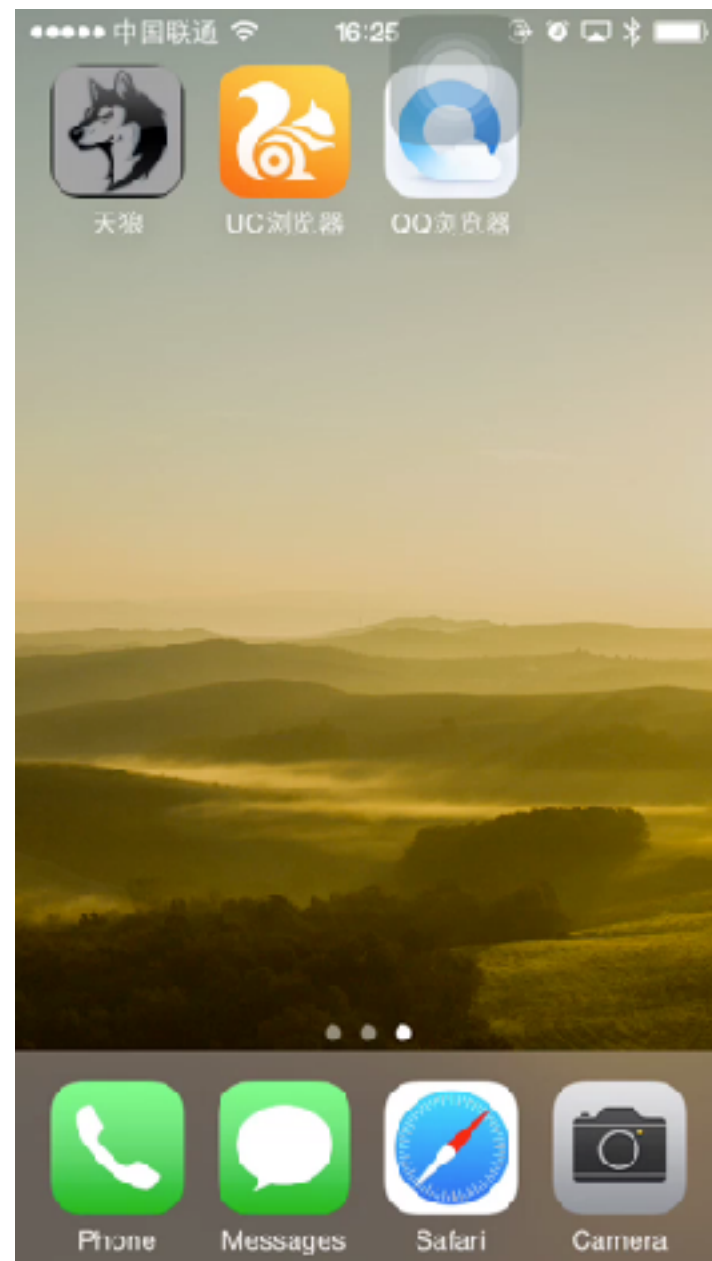
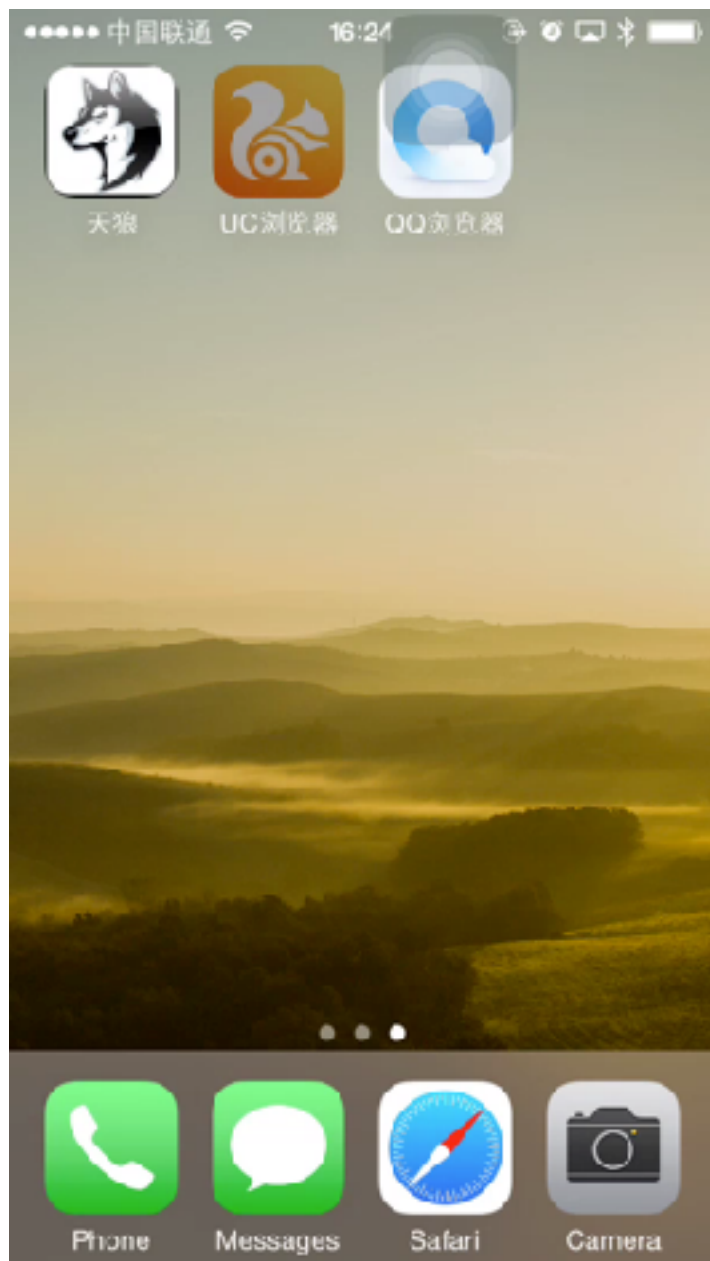




WIFI环境下资源加载耗时



3G环境下资源加载耗时



资源问题其它解决方案:

- 应用缓存 (HTML5 Application Cache) (被新的 Web 标准废弃)
- Cache (Native) (第一次加载体验如何保障)

场景1：我修复了一个Bug，要上线

- 前端界面展示的兼容性出问题了，修改了一个 CSS 文件，用了2行代码
- 配合接口修改一个逻辑，改了10行代码
- PM 说：要发版

But，我的离线包有 1Mb 的大小

local assets > build > local.zip > https://cdn1.url.zip

Do something more intelligent

Config File Management Sys.



Interceptor

The idea is simple, the scene is complex

```
local assets > build > local.zip > https://cdn1.url.zip
```

White list, low cost access

Incremental Calc

Auto push

Config File Management Sys.

场景2：推包频率高

- 一个活动页面，资源位调整，一天发布5次
- 暑期大促，整点开抢，流量大并发高，时效性要求高

思考：离线包推包触达率

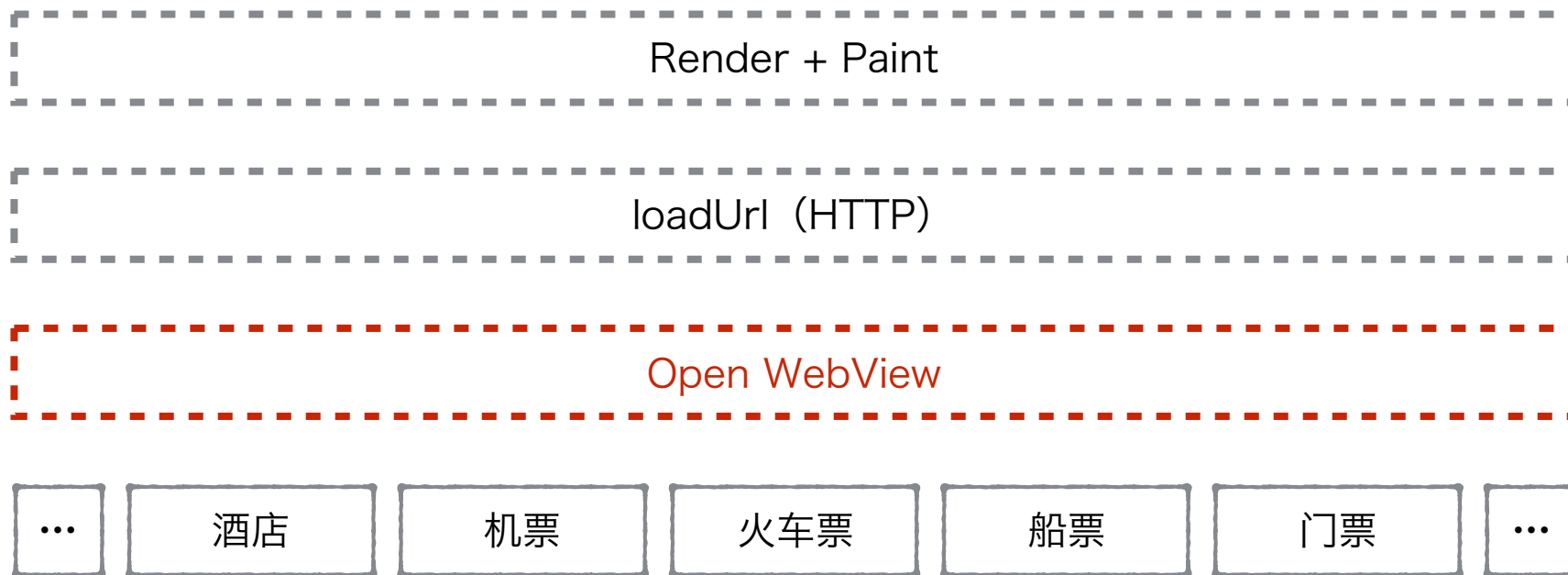
- 推送开启
- 预热，定时切换
- 走线上

场景3：APP体积限制 VS 离线包内置

- 频道内加载（收益变小）
- 工程问题，工程化处理

场景4：离线包文件安全

- md5计算比对文件内容
- cache-info.json 将对应关系存储到内存，减少IO



Open Webview

- 前端摆脱对 webview 的依赖（废话…）
- 并行改为串行
- 预载

Render + Paint

loadUrl (HTTP)

Open WebView

... 酒店 机票 火车票 船票 门票 ...



Render + Paint

- 视觉的head和body分开载入，或者引入其他动效解决视觉上白屏
- css前置 js后置，确保不要阻塞渲染

如何解决类似基于 LBS 场景的原生 API 调用？

- 桥协议 （桥协议是客户端这个宿主为前端提供的拓展 API)

- 工能类: request、sendSms、pay、login...
- 工具类: HBNBVersion、supportApis、installed...
- 信息类: getCity、getUuid...
- ui 类: open、modal、toast...
- 业务类: flight、train、travel、tower...

桥协议还能怎么用？

- 业务闭环
 - H5引流 → Native下单
- 服务闭环
 - 机场信息 → 短信通知 → Native查看行程
 - 预定酒店 → 自动导航

对 Hybrid 的规划

- 更加自动化、工程化的离线包接入体验，降低接入成本；数据收益可视化；可以离线活动页面；提升推包到达率。
- 更加完善的桥协议，KNB 共建，打通从引流到交易到服务的链路闭环，体验为王
- HTTP 请求的移动性能：SPDY（已废弃）、HTTP2

思考: Hybrid 会被终结吗?

- vue + Weex: Write once, run everywhere
- reactNative: Learn once, write anywhere
- reactMix

Q & A



[maquan\[x\]meituan.com](http://maquan[x]meituan.com)